IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 6, June 2016

Algorithm for a Service Provider Recommendation Engine

[1]Aparajita Singh, [2]Piyush Kumar Srivastava, [3]Mr. G. Ramesh
The National Institute of Engineering, Mysore

***Abstract* - Recommendation Engines are common nowadays on various e-commerce websites. These algorithms suggest users with a list of various items they might be interested in. With the growth in e-commerce, the accuracy of these algorithms has been increasing daily. In this paper, we study various approaches for developing such algorithms and analysing them to design our own recommendation algorithm. The final algorithm is based on the approaches discussed in the paper. The algorithm is improved with the help of a survey which was conducted online.**

***Index Terms*—recommendation engine, service provider, energy spreading, graph based recommendation.**

I. INTRODUCTION

In today's world, many of the products and services we need in our daily lives are available online, just a few clicks away. Most of the times, the websites or the applications, providing these services, have some algorithm incorporated in them that suggests various other similar services to the user visiting the website or the application. These algorithms need to be more or less accurate in suggesting the services to the user in order to both, help the user look for services, and also to bring the website or the application more business.

Recommendation engines analyse the taste, the mood or the context in which the user is at the moment. Based on the analysis, they create an accurate recommendation that suits the particular user. There are various techniques used to create recommendations. The two of the main categories of recommendation engines are content based and collaborative. In both of these categories the entire entity and its features are studied to include it in the recommendation. In collaborative approach this represents matrix of users and items, in content-based it is matrix of items and their similarity. However, there is a possibility that the recommendation may still be inaccurate. Many recommendation systems try to recommend item by pairing the extracted knowledge base with the user's context and taste. As a result, the recommender systems may suffer from performance issues what makes them unusable in real time.

The main categories of recommendation engine are discussed below.

1. Content-Based Filtering

Content Based Filtering (CBF) is one of the most commonly used and studied recommendation approaches. In this approach, the interests of the user model are based on inferences from the items that the user interacted with. "Items" are generally in text form and "interactions" are established through actions such as clicks, downloads, etc. Items are represented based on a content model having the item's features. These features are usually words, phrases or n-grams. It is common for these features to be weighted. Once the most discriminating features are identified they are stored in the form of a matrix, usually consisting of the features and weights. The user model will hence consist of a matrix of the features of a user's items.

CBF has several advantages over the stereotyping approach. CBF uses user based personalization and so accounts for the exceptions to stereotypes as well. Also, CBF requires relatively less manual intervention for classification since it can be automated. On the other hand, CBF requires more computing power than stereotyping, thereby creating a performance overhead. Another setback of CBF is that its low serendipity, thereby leading to overspecialization and hence causing the filter bubble [3].

2. *Collaborative filtering*

The theory behind collaborative filtering is that like-minded users typically tend to make similar choices. Based on this premise, when like-minded users were identified, items which received positive reviews from one set of users are presented to their counterparts and vice versa.

Collaborative filtering accounts for the complexity involved in human choices by altogether eliminating the need for detangling the logic behind these choices. This approach abstracts the whole process of decision-making that the user followed and entirely hinges its own choices on the like-minded-ness of the users.

The effectiveness of collaborative filtering is heavily dependent on the ratio between the number of users and the amount of content that recommendations can be drawn from. As mentioned in [2], collaborative filtering works well for a system where the number of users is much higher than the amount of content, but can fail when the ratio is inverted.

3. Graph based recommendation

This approach utilizes the inherent connections between items and then forms a graph based on these items and connections. Once the graph is built, graph metrics such as path length, number of connected nodes, etc. can be used to recommend items

4. Global Relevance

This approach adopts a one-fits-all approach and does not make recommendations specific to the user. Instead, recommendations made are the same for all users and are based on a set of parameters that are common for all users such as overall popularity.

In this paper, we try to develop an algorithm that tries to accurately recommend the service providers from a database based on various parameters. The algorithm should be accurate, but more so, it should be operable in real time, since the size of the database can be huge, containing details of up to thousands of service providers. The algorithm will be based on the discussed approaches to developing such algorithms, discussed in the later section.

II. RECOMMENDATION ALGORITHMS

The recommendation algorithms discussed here are all based on a graph approach and global relevance. The algorithms use a node in the graph to represent an entity and the links between the nodes to represent the attributes of the entity. The graph algorithms make it easy for us to visualise the flow of the algorithms and predict their accuracy. The algorithms discussed try to find the nodes which are the closest in the graph to all initial nodes, which are then returned as recommendations. The recommendations made are not specific to the user. All users in the same geographic location will receive the same recommendations, thereby making the algorithm globally relevant

Due to the above techniques, the algorithms become very versatile and flexible, and can be used in various scenarios.

A. Union Colours Algorithm

The Union Colours Algorithm is based on the *Breadth First Search (BFS)* graph algorithm. According to BFS, we start from the root node, go down one level, visit all the nodes on that level, and then proceed to the next level. The following steps are followed during the application of the algorithm [1]:

1. Mark each node as a different colour (shown as numbers here)
2. Perform a simultaneous BFS from each initial node:
3. Enqueue all initial nodes.
4. Dequeue a node and visit it.
5. Add all neighbours of the visited which are yet to be visited node into the queue.
6. Repeat from step 2.
7. When a node is visited:

i. If it is yet to be coloured, colour it with the colour of the initial node

ii. If it is already coloured (with a different colour), merge the two colours into one - remember that one colour equals the other

8. Keep merging colours until the last two colours are merged and only one colour remains
9. Return the current and the next required number of nodes in the queue as the result

Figure 1 shows how the algorithm is applied to the service provider recommendation engine. Nodes which are of the final colours, formed by merging of most colours in the graph, are recommended.

B. Energy Spreading Algorithm

The Energy Spreading Algorithm is based on *Spreading Activation*, which involves the entities having some specific energy in the start, which spreads through the graph while traversals.

IFERP
connecting engineers... developing research
ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 6, June 2016

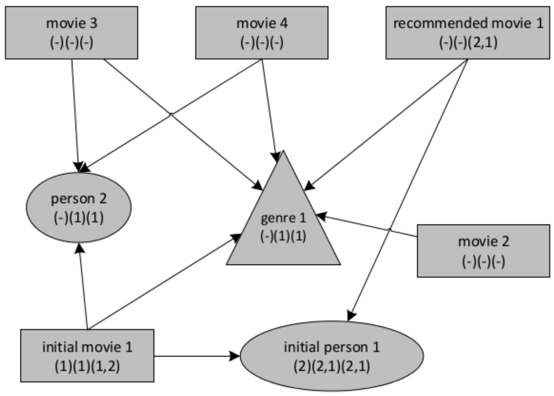

Fig 1: Union Colours algorithm

Image courtesy: Movie Recommendation Based on Graph Traversal Algorithms - Lubos Demovic, Eduard Fritscher, Jakub Kriz, Ondrej Kuzmik, Ondrej Proksa, Diana Vandlikova, Dusan Zelenik, Maria Bielikova

The algorithm is implemented as follows [1]:

1. Set the energy of each initial node to some constant value.
2. Perform a simultaneous BFS from each initial node.
3. When a node is visited its energy increases by value E, E = Ep, where Ep is the energy of the parent node which enqueued the visited node and n is the number of nodes the parent node enqueued.
4. A node's energy can increase multiple times, but it only spreads it when it receives energy for the first time.
5. Continue until the required number of nodes is visited from each initial node.
6. Order the nodes by their energies and return the required number of the nodes, the more energy it has the higher it is.

Figure 2 shows how the algorithm is implemented to a recommendation engine in order to accurately suggest items.

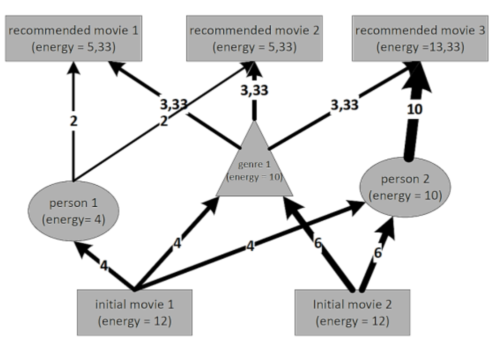

Fig 2: Energy Spreading Algorithm

Image courtesy: Movie Recommendation Based on Graph Traversal Algorithms - Lubos Demovic, Eduard Fritscher, Jakub Kriz, Ondrej Kuzmik, Ondrej Proksa, Diana Vandlikova, Dusan Zelenik, Maria Bielikova

III. SUGGESTED ALGORITHM

The algorithm we use here is a combination of both Union Colours Algorithm and the Energy Spreading Algorithm, discussed in the previous section. The Union Colours Algorithm links each of the start entities to the final items, while the energy spreading algorithm can have priorities on various parameters considered while recommending the items.

Here, we develop a recommendation algorithm for an Android application which suggests service providers such as doctors, lawyers, engineers, etc. to the user. The algorithm should be able to recommend the most suitable service providers to the users based on specific parameters. The parameters we consider here are:

7. Rating of the service providers, calculated from the reviews of various users.
8. Distance of the service providers from the user's location

We have chosen these parameters as an example here. The algorithm can be extended and applied to various other parameters like cost of the service, reliability, availability, etc.

The algorithm uses a survey that helps in deciding the starting energies of the parameters. The survey is explained in detail in the next section.

The energy for each service provider entry is computed by the following formula:

$$E_{final} = E_{distance}.W_{distance} + E_{rating}.W_{rating} \quad (1)$$

, *where, Edistance* is the energy corresponding to the distance between the user and the service provider, *Erating* is the energy corresponding to the rating of the service provider, *W* is the weight of the entry amongst other entries.

An entry is assigned a value of either 3, 2 or 1 based on the distance between the user and the service provider, called *d*. For a city of diameter around 10 km, the *d* value is 3 for distance up to 2 km, 2 for 2 km to 5 km and 1 for values greater than 5 km. This value is used to calculate *Wdistance* which is calculated by the following formula:

$$W_{distance} = \frac{d}{\Sigma d} \quad (2)$$

Similarly,

IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 6, June 2016

$$W_{rating} = \frac{r}{\Sigma r} \quad (3)$$

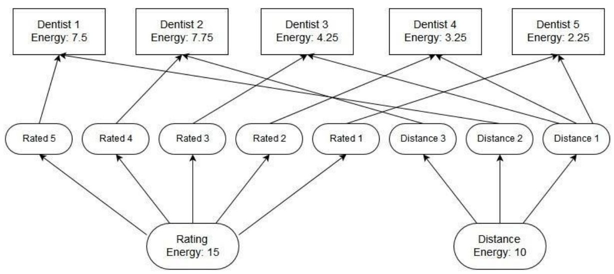

Fig.3: Recommendation Algorithm for Service Providers

Here the energies are calculated in the following manner. Let us consider Dentist 1. *d* is 2, Σ*d* is 8, therefore, *Wdistance* is 0.25. *r* is 5, Σ*r* is 15, therefore, *Wrating* is 0.33. *Erating* is taken as 15, and *Edistance* is taken as 10 here. Hence, *Efinal* is calculated as 7.5.

IV. EXPERIMENTAL EVALUATION

Edistance and Erating are calculated based on a survey. The survey contained 15 entries each corresponding to one of the combination of 5 ratings and 3 distances. The survey is shown below:

Number the following service providers, dentists here, in order of your preference:

1. *Dentist, rated 4/5, 4 km away*
2. *Dentist, rated 3/5, 8 km away*
3. *Dentist, rated 2/5, 2 km away*
4. *Dentist, rated 5/5, 6 km away*
5. *Dentist, rated 1/5, 1 km away*
6. *Dentist, rated 5/5, 5 km away*
7. *Dentist, rated 3/5, 2 km away*
8. *Dentist, rated 1/5, 6 km away*
9. *Dentist, rated 2/5, 4 km away*
10. *Dentist, rated 3/5, 4 km away*
11. *Dentist, rated 4/5, 1 km away*
12. *Dentist, rated 4/5, 8 km away*
13. *Dentist, rated 5/5, 2 km away*
14. *Dentist, rated 1/5, 3 km away*
15. *Dentist, rated 2/5, 8 km away*

The preferences are given values in the opposite order of their selection, i.e. the values are 15, 14, 13...1 for preferences 1, 2, 3...15 respectively.
Erating and Edistance are calculated as:

$$E_{rating} = \frac{\Sigma vi.ri}{\Sigma ri} \quad (4)$$

where, vi is the value of the preference and ri is the rating of the preference.

$$E_{distance} = \frac{\Sigma vi.di}{\Sigma di} \quad (5)$$

where, vi is the value of the preference and di is the rating of the preference.

From the survey the values for *Erating* is calculated to be *9.63* and that of *Edistance* is *8.54*. These values can be used to implement this algorithm in the Service Provider Recommendation Engine.

The data from the survey is as follows:
No. of people involved: 20
Σri = 20 x (1+2+3+4+5) x 3 = 900
Σvi.ri = 8670
∴ Erating = 8670/900 = 9.63
Σdi = 20 x (1+2+3) x 5 = 600
Σvi.di = 5126
∴ Edistance = 5126/600 =8.54

V. CONCLUSION

In this paper we developed a graph based recommendation algorithm with global relevance after studying various approaches for recommendation algorithm. Our algorithm can be applied to most of the e-commerce applications and websites, since it is capable of recommending items in real time. The main advantage of this algorithm is that the parameters that are important for recommending items to the users don't have to same as what we have demonstrated here. Any parameter can be included in the algorithm as long as it has been given an appropriate energy value.

The algorithm can be further improved by including various machine learning techniques to calculate the starting energies of the parameters instead of carrying out a survey, thereby making the algorithm self-sufficient.

REFERENCES

1. Movie Recommendation Based on Graph Traversal Algorithms Luboˇs Demoviˇc, Eduard Fritscher, Jakub Kˇr´ıˇz, Ondrej Kuzm´ık, Ondrej Proksa, Diana Vandl´ıkov´a, Duˇsan Zelen´ık, M´aria Bielikov´a, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology, Ilkoviova, 842 16, Bratislava, Slovakia televido@googlegroups.com

2. Research-Paper Recommender Systems: A Literature Survey Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger

3. Exploring the Filter Bubble: The Effect of Using Recommender Systems on Content Diversity Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen and Joseph A. Konstan